Apriori 알고리즘에 의한 연관 단어 지식 베이스에 기반한 가중치가 부여된 베이지안 자동 문서 분류

고수정[†] · 이정현[††]

요 약

기존의 베이지안 문서 분류를 위한 단어 군집 방법은 많은 시간과 노력을 요구하며, 단어 간의 의미 관계를 정확하게 반영하지 못하는 문제점이 있다. 본 논문에서는 마이닝 기법으로 구축된 연관 단어 지식 베이스를 기반으로 하는 베이지안 문서 분류 방법을 제안한다. 제안된 베이지안 문서 분류 방법은 문서를 분류하기 전에 훈련 문서를 사용하여 가중치가 부여된 연관 단어 지식 베이스를 구축한다. 그 다음으로, 베이지안 확률을 이용하는 분류자는 구축된 연관 단어 지식 베이스를 기반으로 문서를 클래스별로 분류한다. 제안된 방법의 성능을 평가하기 위해, 상호 정보 계산에 의한 단어 사전을 이용한 가중치가 부여된 베이지안 문서 분류 방법, 가중치가 부여된 베이지안 분류 방법, 기존의 단순 베이지안 분류 방법과 비교하였다. 그 결과, 연관 단어 지식 베이스에 기반한 가중치가 부여된 베이지안 분류 방법이 상호 정보에 의한 단어 사전을 이용하는 가중치가 부여된 베이지안 분류 방법보다는 0.87%, 가중치가 부여된 베이지안 분류 방법보다는 2.77%, 단순 베이지안 방법보다는 5.09% 높은 성능 차이를 보였다.

Weighted Bayesian Automatic Document Categorization Based on Association Word Knowledge Base by Apriori Algorithm

Ko, SooJung[†] and Lee, JungHyun[††]

ABSTRACT

The previous Bayesian document categorization method has problems that it requires a lot of time and effort in word clustering and it hardly reflects the semantic information between words. In this paper, we propose a weighted Bayesian document categorizing method based on association word knowledge base acquired by mining technique. The proposed method constructs weighted association word knowledge base using documents in training set. Then, classifier using Bayesian probability categorizes documents based on the constructed association word knowledge base. In order to evaluate performance of the proposed method, we compare our experimental results with those of weighted Bayesian document categorizing method using vocabulary dictionary by mutual information, weighted Bayesian document categorizing method, and simple Bayesian document categorizing method. The experimental result shows that weighted Bayesian categorizing method using association word knowledge base has improved performance 0.87% and 2.77% and 5.09% over weighted Bayesian categorizing method using vocabulary dictionary by mutual information and weighted Bayesian method and simple Bayesian method, respectively.

1. 서 론

인터넷의 인기가 높아지면서 웹문서와 이를 사용하는 사람들의 수가 점차로 증가되었다. 이에 따라 정보 검색을 효율적으로 하기 위하여 웹문서를 자동으로 분류하려는 여러 방법이 연구되어 왔다[17]. 문서의 자동 분류에 대한 기존의 연구는 확률을 이용한 방법[9,14], 통계적인 기법을 이용한 방법[3,6], 벡터 유사도를 이용하는 방법[15], 엔트로피를 이용하는 방법[15] 등이 있다. 이들 중에서 확률을 이용하여 학

[†] 정회원, 인하대학교 대학원 전자계산공학과
[††] 인하대학교 전자계산학과 교수

습하는 문서 분류 방법이 가장 많이 연구되었으며, 이 방법은 일반적인 문서 집합에 대해 높은 분류 효율을 나타내고 있다[14].

본 논문에서는 Apriori 알고리즘에 의한 연관 단어 지식 베이스의 카테고리를 기반으로 문서를 분류하는 가중치가 부여된 베이지안 문서 분류 방법을 제안한다. 기존의 단순 Naive Bayes[13]를 사용한 분류는 문서에 출현한 모든 단어에 대해서 추정치를 계산하고 이를 바탕으로 분류를 수행하였기 때문에 문서의 특징을 정확히 반영하기 어렵고, 많은 잡음들의 영향으로 문서를 오분류하게 된다. 이를 개선한 가중치가 부여된 베이지안 문서 분류 방법[18]은 각 문서 내의 모든 단어를 특징으로 사용하는 것이 아니라 문서 내의 단어에 대해 가중치를 계산하고 가중치가 높은 단어만을 특징으로 추출한다. 또한 추출된 단어의 수가 작을 경우 상호 정보를 이용한 단어 군집으로 특징에 사용될 단어의 수를 증가시킨다. 제안된 방법은 기존의 Naive Bayes에 의한 분류보다는 정확도를 높였으나 특징으로 추출된 단어가 단어 간의 의미 관계를 반영하지 못하므로 단어의 의미 중의성 문제를 해결하지 못하였다. 이를 해결하기 위해, 본 논문에서 제안한 특징 추출 방법은 마이닝 기법이다[16]. 마이닝 기법은 단어 간의 의미 관계가 고려되도록 문서로부터 연관 단어를 추출한다.

본 논문에서 제안한 베이지안 문서 분류 방법은 문서를 분류하기 위해 먼저 연관 단어 지식 베이스를 구축한다. 다음으로, 이러한 지식 베이스의 연관 단어를 대상으로 Naive Bayes 학습을 함으로써 가중치를 부여한다. 마지막으로, Naive Bayes 분류자는 가중치가 부여된 연관 단어 지식 베이스의 클래스의 하나로 문서를 분류한다. 이러한 방법으로 제안된 방법의 성능을 평가하기 위해, 상호 정보 계산에 의한 단어 사전을 이용한 가중치가 부여된 베이지안 문서 분류 방법, 가중치가 부여된 베이지안 분류 방법, 기존의 단순 베이지안 분류 방법과 비교하였다.

2. 관련 연구

2.1 문서 분류

텍스트 문서의 분류를 위한 대부분의 연구[7,11]는 Naive Bayes 분류자라고 불리는 변형된 베이지안 분류법을 사용하였다. Joachims[4]은 베이지안 분류자를 사용하여 유즈넷 뉴스 기사 분류를 시도하여, 그 결과로 89%의 분류 정확도를 얻을 수 있음을 보였다. Lang[8]은 순위화된 기사를 학습 집합으로 사용하여 사용자가 관심이 있는 기사를 예측함을 보였다. 또한, Lewis[10]는 Maron[12]이 사용한 통계적인 방법을 이용하는 텍스트 분류 방법과 베이지안 분류식을 사용하여 Reuters-22173집합에 대해 분류를 실험한 결과, 베이지안 분류식을 적용한 분류 방법에서 더 좋은 분류 효율을 얻을 수 있음을 보였다.

McCallum[14]은 기존의 Naive Bayes 가정을 사용한 연구들을 크게 두 가지의 형태로 분류하여 비교하고, 그들의 성능을 실험을 통하여 비교하였다. 첫 번째 형태는 문서 내의 단어들의 발생과 비발생만을 고려하여 문서를 분류하는 방법으로, 일반적으로 이진 독립 모델(Binary Independence Model)이라 칭하거나 특별히 문서 분류에 있어서 다중 이형 베르눌리 모델(Multi-variate Bernoulli Model)이라고도 한다. 두 번째 형태는 문서 내의 단어의 발생과 비발생뿐만 아니라 해당 단어의 출현 빈도까지 고려하는 방법으로 일반적으로 다항 모델(multinomial Model)이라 부른다. McCallum은 위의 두 가지 연구를 토대로 웹문서, 유즈넷 기사, Reuters newwire기사를 포함하는 다섯 개의 문서 집합에 대해 두 가지 분류 방법을 적용하여 분류 효율을 비교하였다. 그 결과, 다항 모델이 다중 이형 베르눌리 모델에 비해 평균 27%의 에러가 감소됨을 보였다.

본 논문에서는 학습 문서들로부터 사전 확률 값을 계산하기 위해 단어의 발생 여부를 사용하는 방법이 아닌 단어의 출현 빈도를 고려하는 다항(multinomial) 베이지안 학습법을 사용한다[13].

2.2 Apriori 알고리즘

연관 규칙은 한 항목들의 그룹과 다른 항목들의 그룹 간에 강한 연관성이 있음을 밝혀 준다. 예를 들면, 소매점에서 각 고객이 구매하는 물품들의 집합을 한 트랜잭션이라 하고, 이런 트랜잭션들을 일정한 기간동안 저장한 것을 데이터베이스라 하면, 기저귀를 사는 사람은 맥주를 구매한다는 것을 규칙으로 표현하면, 기저귀=>맥주[10%의 support]와 같이 표현할 수 있다. 여기서 10%의 지지도(support)라는 것은 주어진 데이터베이스의 트랜잭션(고객들) 중에서 10%가 기저귀와 맥주를 동시에 산다는 것이고, 80%의

신뢰도(conference)라는 것은 기저귀를 사는 고객들 중에서 80%가 맥주를 산다는 것이다. 연관 규칙 탐사에서는 사용자가 지지도와 신뢰도의 값을 적절하게 입력함으로써 이미 발생한 트랜잭션들에서 물품들 상호 간의 연관성을 발견해낼 수 있다[1].

연관 규칙 마이닝 알고리즘인 Apriori는 구매하는 물품들의 집합인 트랜잭션으로부터 연관 규칙을 마이닝한다. 연관 규칙은 두 단계를 통하여 구성된다[2]. 첫번째 단계는 최소의 지지도(min_support) 이상의 발생 지지도(transaction support)를 가지는 조합을 찾아 빈발 단어 항목을 구성한다. 두번째 단계는 데이터베이스로부터 연관 규칙을 생성하기 위하여 빈발 항목 집합을 사용한다. 모든 빈발 항목 집합(L)에 대해서 빈발 항목 집합의 모든 공집합이 아닌 부분집합들을 찾는다. 각각의 그러한 부분집합(A)에 대하여, 만약 support(A)에 대한 support(L)의 비율이 적어도 최소 신뢰도(min_conference) 이상이면, A->(L-A)의 형태의 규칙을 출력한다. 이 규칙의 지지도는 support(L)이고, 신뢰도는 support(L)/support (A)이다. Apriori 알고리즘에서 후보집합의 생성은 Apriori-gen을 사용하여 새로운 후보집합을 만들게 함으로써, 후보항목의 수를 줄일 수 있다. 이에 따라 연관 규칙을 찾는 시간이 감소된다. 연관 규칙을 찾는 Apriori 알고리즘은 그림 1과 같다.

```
L1:={large 1-itemsets};   //빈발 항목을 구성
For (k=2:Lk-1 <> 0; k++) do begin
  Ck=Apriori-gen(Lk-1); // 새로운 후보항목
   Forall transactions t ∈ D do begin
     Ct=subset(Ck,t);//  t 에 포함된 후보항목
     Forall candidates c ∈ Ct do
      c.count++;
   end
   Lk={c ∈ Ck|c.count >= min_support} // 최소
   지지도 이상의 항목의 조합을 추출
End
Answer=UkLk;
```

그림 1. 연관 규칙을 찾는 Apriori 알고리즘

3. 연관 단어 지식 베이스에 기반한 베이지안 자동 문서 분류

3.1 문서의 특징 표현

본 논문에서는 텍스트로 이루어진 문서를 표현하기 위해 형태소 분석을 통한 명사 추출 과정을 전처리 과정으로 사용한다. 전처리 과정을 통하여 추출된 명사들을 대상으로 연관 단어를 마이닝함으로써 각 문서를 연관 단어들의 집합, 즉 연관 단어 벡터 모델로 나타낸다.

전처리 과정의 형태소 분석이란 하나 또는 둘 이상의 형태소로 이루어진 단어에 대하여 단어를 이루고 있는 형태소를 분리한 후에 형태론적 변형이 일어난 형태소의 원형을 복원하고 사전과 단어 사이의 통합 관계에 대해 옳은 분석 후보를 선택하는 과정으로 구성된다. 연관 단어 벡터 모델은 형태소 분석의 복잡한 부분인 파싱(parsing)을 통한 의미 분석을 생략하고 추출된 명사만을 사용한다.

그림 1의 Apriori 알고리즘은 형태소 분석에 의해 추출된 명사들로부터 연관 단어를 마이닝한다. 그 결과, 문서는 $\{(w_{11}\&w_{12}\&\cdots\&w_{1(a1-1)}=>w_{1a1}), (w_{21}\&w_{22}\&\cdots\&w_{2(a2-1)}=>w_{2a2}), \cdots, (w_{k1}\&w_{k2}\&\&w_{k(ak-1)}=>w_{kak}), \cdots, (w_{p1}\&w_{p2}\&\cdots\&w_{p(ap-1)}=>w_{pap})\}$ 형태의 연관 단어 벡터 모델로 표현된다. 여기서, $(w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak})$ 등의 형태는 연관 단어를 나타낸다. 이러한 형태 안의 "&" 기호는 단어와 단어가 연관되었음을 나타내는 기호이다. 또한, $\{w_{p1}, w_{p2}, w_{p(ap-1)}, \cdots, w_{pap}\}$는 연관 단어를 구성하는 단어들의 구성이며, ap은 연관 단어를 구성하는 단어의 수이고, p는 문서를 대표하는 연관 단어의 수이다.

3.2 연관 단어 지식 베이스을 위한 연관 단어의 마이닝

그림 1의 Apriori 알고리즘은 데이터 마이닝 기법을 이용하여 단어 간의 연관 규칙을 마이닝한다. 이 경우, 사용되는 데이터베이스는 문서의 이름과 문서에서 추출된 명사들로 구성되는 데, 알고리즘에 사용되는 빈발 단어 항목과 후보 단어 항목은 문서를 대상으로 형태소 분석을 통해 추출된 명사이다.

문서에서 추출된 명사를 Apriroi 알고리즘에 적용하여 연관 단어 쌍을 구성하기 위해서는 신뢰도와 지지도를 결정해야 한다. 2.2절에서 기술한 바와 같이 신뢰도와 지지도를 어떻게 지정하는 가에 따라 마이닝되는 연관 규칙의 수와 내용에서는 많은 차이를 보인다. 따라서, 연관 단어 지식 베이스를 구축하기에 적합한 신뢰도와 지지도를 지정해야만 지식 베이스에 포함될 연관 단어가 적합하게 마이닝된다.

신뢰도를 결정하기 위한 식 (1)은 단어가 한 문서 내에서 공기한 정보를 나타낸다.

$$Confidence(w1->w2)=Pr(w2|w1) \quad (1)$$

그림 2는 100개의 문서를 대상으로 신뢰도를 다양하게 변화시켰을 때, 마이닝되는 연관 단어에 대한 정확도와 재현율을 나타낸다. 마이닝된 결과에 대해 재현율과 정확도를 평가하는 기준은 영어 단어에 대한 시소러스인 WordNet을 사용하여 평가하였다. 단어들을 의미에 따라 영어 단어로 번역하여 WordNet으로 서로 비교하였을 때, 다른 단어들과 의미가 유사하지 않은 단어들로 연관 단어가 구성되었을 경우 오류로 처리했다.
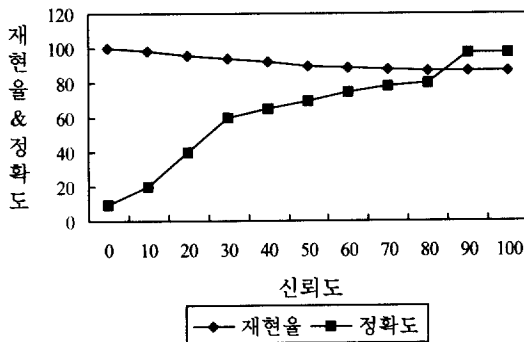
그림 2. 신뢰도의 변화에 따른 재현율과 정확도

위의 그림은 신뢰도가 클수록 마이닝되는 연관 단어의 정확도는 높아지나 재현율은 낮아짐을 나타낸다. 그러나 85이상의 신뢰도에서는 재현율이 거의 일정하고 정확도는 높은 수치를 나타낸다. 따라서 가장 적합한 연관 단어를 마이닝하기 위해서는 신뢰도를 85이상으로 지정해야 한다.

지지도를 결정하기 위한 식 (2)은 전체 단어들의 쌍 중에 각 연관 단어의 출현 빈도를 나타낸다. 지지도가 크다면 빈도수는 작으나 중요한 연관 단어가 생략될 수 있고, {기본&방식&이용&지정=>실행}과 같이 빈도수는 높지만 중요하지 않은 연관 단어가 마이닝된다.

$$Support(w1->w2)=Pr(w1 \cup w2) \quad (2)$$

그림 3은 100개의 문서를 대상으로 지지도를 다양하게 변경시킴에 따른 정확도와 재현율의 변화를 나타낸다. 마이닝된 결과에 대해 재현율과 정확도를 평가하는 기준은 신뢰도와 같이 영어 단어에 대한 시소러스인 WordNet을 사용하여 평가하였다.
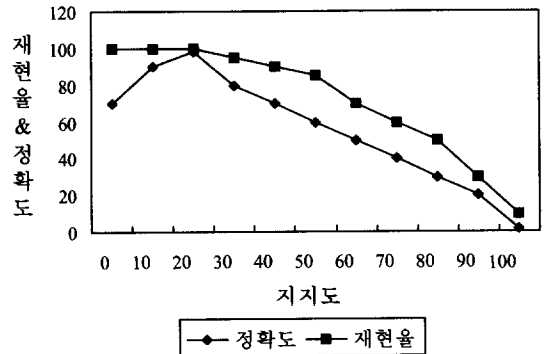
그림 3. 지지도의 변화에 따른 재현율과 정확도

정확도와 재현율의 곡선이 일치하는 지점은 지지도가 22인 경우로, 이 지점에서 가장 적합한 연관 단어가 마이닝된다. 그러나, 지지도가 22이상인 경우에는 정확도와 재현율이 모두 낮아진다. 따라서 가장 신뢰할 만한 연관 단어를 추출하기 위해서는 22이하의 지지도를 지정해야 한다. 그러나 지지도를 0으로 한다면 클래스와 관계가 없는 문서에서 연관 단어가 추출되므로 0보다 크도록 설정하여야 한다.

클래스별로 마이닝된 연관 단어는 연관 단어 지식 베이스에 저장된다. 연관 단어 지식 베이스는 $\{class_1, class_2, \cdots, class_{ID}, \cdots, class_N\}$의 클래스로 구성되며, $\{class_1, class_2, \cdots, class_{ID}, \cdots, class_N\}$는 연관 단어 지식 베이스의 클래스의 레이블을 의미한다. 각 클래스는 같은 구조를 갖기 때문에 그림 4에서는 $class_{ID}$의 구조만을 보인다. 그림 4에서 $\{w_{m1}\&w_{m2}\&\cdots\&w_{m(am-1)}=>w_{mam}\}$에서 m은 각 클래스에 마이닝된 연관 단어들의 총 수를 의미하며, am은 하나의 연관 단어를 구성하기 위한 단어의 수를 의미한다. 여기서, m의 값은 클래스마다 다르게 지정될 수 있다. 그러한 이유는 같은

$$\begin{aligned} class_{ID}:\{ & (w_{11}\&w_{12}\&\cdots\&w_{1(a1-1)}=>w_{1a1}), \\ & (w_{21}\&w_{22}\&\cdots\&w_{2(a2-1)}=>w_{2a2}), \\ & \cdots, \\ & (w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak}), \\ & \cdots, \\ & (w_{m1}\&w_{m2}\&\cdots\&w_{m(am-1)}=>w_{mam})\} \end{aligned}$$

그림 4. 연관 단어 지식 베이스의 구조

훈련 문서를 대상으로 Apriori 알고리즘을 실행한 결과, 마이닝된 연관 단어의 수는 다르게 나타나기 때문이다. 또한 am도 연관 단어에 따라 다른 값을 보인다. 훈련 문서의 내용이 다르기 때문에 연관 단어를 구성하는 단어의 수는 각기 다른 값을 갖는다.

3.3 Naive Bayes 학습에 의한 가중치 부여

Naive Bayes 알고리즘은 학습 단계와 분류 단계를 통하여 문서를 분류할 수 있다. 학습 단계에서는 Apriori 알고리즘에 의해 구축된 연관 단어 지식 베이스의 연관 단어에 가중치를 부여한다. 가중치를 부여하기 위해서 우선 가중치를 부여하기 위한 훈련 문서를 수집한다. 구성된 훈련 문서로부터 3.2절에서 설명한 방법으로 연관 단어를 마이닝한다. 마이닝된 연관 단어는 다른 문서에 나타난 연관 단어에 관계없이 독립적이라고 가정한다. 이러한 가정에서 지식 베이스의 $class_{ID}$에 있는 k번째 연관 단어 ($w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak}$)로 가중치를 부여하기 위해서 식 (3)을 이용한다. 본 논문에서는 $class_{ID}$에 있는 k번째 연관 단어 ($w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak}$)의 가중치는 $P((w_{k1}\&w_{k2}\&w_{k(ak-1)}=>w_{kak})|class_{ID})$로, $class_{ID}$에서의 ($w_{k1}\&w_{k2}\&w_{k(ak-1)}=>w_{kak}$) 출현 확률로 표현한다. 여기서, n은 훈련 문서의 101번째부터 200번까지의 문서로부터 마이닝된 연관 단어의 전체 수이고, n_k는 전체 개수 n 중에서 지식 베이스에 있는 연관 단어 ($w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak}$)와 일치하는 연관 단어의 수이다. 또한, $class_{ID}$는 연관 단어 지식 베이스에 있는 클래스의 레이블이며, |AWKB|는 클래스별로 분류된 훈련 문서의 첫 번째 문서로부터 100개까지 구성된 문서로부터 마이닝된 연관 단어를 대상으로 구축된 연관 단어 지식 베이스에 있는 전체 연관 단어의 수이다. 식 (3)의 분모에는 훈련 문서 1부터 200번까지의 정보를 모두 표현함으로써 정확도를 높이기 위하여 |AWKB|을 부가하였다. 또한 분자에는 n_k에 1을 더하여 확률이 0이 되는 것을 예방하였다.

$$P((w_{k1}\&w_{k2}\ldots\&w_{k(ak-1)}=>w_{kak})|class_{ID})=P(k|class_{ID})=\frac{n_k+1}{n+|AWKB|} \quad (3)$$

학습 과정은 누적 단계와 가중치 부여 단계로 나눈다. 누적 단계에서는 훈련 문서에 있는 연관 단어가 지식 베이스 안에 있는 경우 횟수를 누적시킨다. 가중치 부여 단계에서는 누적 단계의 결과를 식 (3)에 적용하여 지식 베이스의 연관 단어에 가중치를 부여한다. 이러한 과정을 통해, 지식 베이스의 연관 단어에 가중치가 추가된다.

3.4 Naive Bayes 분류자에 의한 문서 분류

분류 단계에서는 가중치가 부여된 연관 단어 지식 베이스를 사용하여 Naive Bayes 분류자에 의해 실험 문서를 클래스로 분류한다. 분류를 위해 실험 문서는 3.1절의 방법과 같이 D={d($w_{11}\&w_{12}\&\cdots\&w_{1(a1-1)}=>w_{1a1}$),d($w_{21}\&w_{22}\&\cdots\&w_{2(a2-1)}=>w_{2a2}$),⋯,d($w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak}$),⋯,d($w_{p1}\&w_{p2}\&\cdots\&w_{p(ap-1)}=>w_{pap}$)}의 연관 단어 형태로 표현된다. 여기서, p는 문서 D를 대표하는 연관 단어의 수이다. d는 실험 문서에서 추출된 연관 단어임을 강조하기 위해 연관 단어 앞에 추가한 것이다. 이러한 이유는 실험 문서와 훈련 문서에서 추출된 연관 단어의 형태가 같기 때문에 이를 구별해야 하기 때문이다. 이와 같은 형태로 연관 단어가 추출되었다면, 식 (4)의 가중치를 고려하는 베이지안 분류자를 통해 확률값이 가장 높은 클래스($class$)에 문서를 할당하게 된다. 식 (4)는 문서에서 추출된 연관 단어들이 클래스 $class_{ID}$에 포함될 확률의 곱을 나타낸다.

$$class = \arg\max_{class_{ID}} P(class_{ID})\prod_{i=1}^{m} P(a_i|class_{ID}) \quad (4)$$

식 (4)에서 문서D가 분류될 클래스는 $class$로, 전체 클래스의 수는 N으로, 가중치가 부여된 연관 단어 지식 베이스의 $class_{ID}$에 있는 연관 단어의 수는 m으로 표현한다. 또한 $P((w_{k1}\&w_{k2}\&\cdots\&w_{k(ak-1)}=>w_{kak})|class_{ID})$는 문서D를 표현하는 연관 단어가 가중치가 부여된 연관 단어 지식 베이스의 $class_{ID}$에 존재할 확률을 표현한다. $P(class_{ID})$는 $class_{ID}$의 출현 확률을 나타낸다.

4. 전체 시스템 설계 및 베이지안 문서 분류의 예

이 장에서는 전체 시스템 설계도에 따라 연관 단어 지식 베이스를 구축하며, 이를 기반으로 문서를 분류하는 방법을 구체적으로 설명한다. 그림 5는 본 논문에서 설계한 베이지안 자동 문서 분류를 위한 시스템 구성도를 나타낸다.
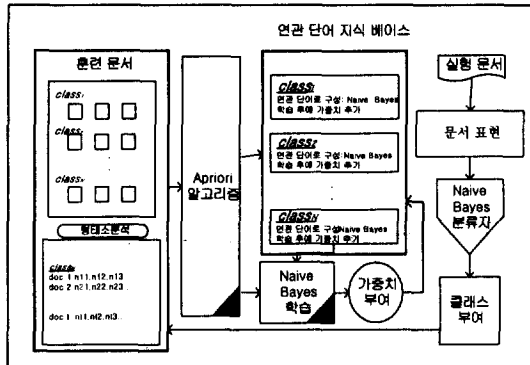
그림 5. 연관 단어 지식 베이스를 기반으로 하는 가중치가 부여된 베이지안 분류자의 구성도

4.1 훈련 문서 및 실험 문서

훈련 문서는 한국어 정보 검색 시스템의 성능 평가용 데이터 집합인 KTset95 문서 4,414개 중 1600개의 문서로, 실험 문서는 웹문서 수집기에 의해 컴퓨터 분야의 URL로부터 수집한 800개의 웹문서와 KTset95 문서 중 800개의 문서를 병합하여 구성한다. 훈련 문서의 클래스는 수작업으로 전산학 각 연구 분야의 8개 클래스로 분류하였다. 훈련 문서와 실험 문서의 실험 대상을 다르게 설정한 이유는 본 논문에서 제시한 방법에 대한 정확한 평가를 위함이다. 여기서 8개의 클래스는 {게임, 그래픽, 뉴스와 미디어, 반도체, 보안, 인터넷, 전자출판, 하드웨어}의 레이블로 표현된다. 이렇게 8개의 클래스로 분류한 기준은 알타비스타, 야후, 한미르 등의 기존의 정보 검색 엔진이 컴퓨터 분야의 주제를 대상으로 분류한 통계에 따른 것이다. 따라서 각 클래스에 200개의 문서가 훈련 문서로 할당된다. KTset95 문서 중 정의된 클래스에 해당하지 않는 문서들은 사용하지 않았다. 그림 5의 훈련 문서에서 {doc 1, doc 2,⋯,doc t}는 훈련을 위해 클래스로 분류한 문서를 의미한다. 한 클래스에 200개의 문서가 속하게 되므로, t는 200의 값을 나타낸다. {nt1, nt2, nt3,⋯}은 문서 doc t를 대상으로 형태소 분석한 결과 추출된 명사를 의미한다.

표 1. 형태소 분석에 의해 추출된 명사의 예

클래스	형태소 분석 결과 추출된 명사들
게임	게임, 경고, 인가, 사용자, 이벤트, 참가...
그래픽	멀티미디어, 출판사, 컴퓨터, 인테리어, 활용
뉴스와 미디어	인터넷, 날씨, 방송, 신문, 환경, 오염,...
반도체	구축, 설계, 창업, 기술, 산업, 메모리,...
보안	해킹, 접근, 발표, 정보, 활동, 인공지능...
인터넷	네트워크, 컴퓨터, 정보,교환, 프로토콜...
전자출판	서점, 결제시스템, 출판, 기획, 제작, 내용...
하드웨어	메인보드, 하드웨어, 하드디스크, 모니터...

4.2 연관 단어 지식 베이스의 구축

연관 단어 지식 베이스를 구축하기 위한 전처리 과정으로서 훈련 문서 중 각 클래스별로 첫 번째부터 100개까지의 문서를 대상으로 형태소 분석을 한다. 그 결과, 표 1과 같은 형태의 명사를 추출할 수 있다.

Apriori 알고리즘은 표 1과 같이 추출된 명사로부터 연관 단어를 마이닝한다. 그 결과는 표 2와 같은 형태로 나타난다. 이러한 자료로 구성된 연관 단어 지식 베이스는 평균 신뢰도 95.3과 평균 지지도 20.1를 나타내며, 총 231개의 연관 단어를 저장한다.

구체적으로, 표 3은 연관 단어 지식 베이스의 8개 클래스 중 게임 클래스에 마이닝된 연관 단어를 보인다.

표 2. 클래스별로 마이닝된 연관 단어

클래스	선행단어(Antecedent)	후행단어(Consequent)	평균 신뢰도	평균 지지도	총 연관 단어 수
게임	*게임&구성&선수&경기&스포츠&참가*	*선 발*	91.30%	20.1039%	18
그래픽	*방법&중심&제작&사용*	*평 가*	88.10%	21.4286%	27
뉴스와 미디어	*뉴스&제공&홍보&속보*	*안 내*	99.9%	20.2838%	30
반도체	시스템&사업&활용&기법	*컴퓨터*	96.20%	20.3839%	22
보안	*세계&네티즌&인물&조작&수법*	*해 커*	95.30%	21.7583%	25
인터넷	*컨텐츠&사이트&관리&쇼핑몰*	*웹*	94.90%	19.3838%	28
전자출판	*입력&편집&출력&컬러&종류*	*출 판*	91.30%	18.2129%	29
하드웨어	*보드&주변기기&슬롯&펜티엄*	*기 기*	90.20%	21.2532%	30

표 3. 연관 단어 지식 베이스의 연관 단어(게임 클래스)

(1) 게임&구성&선수&경기&스포츠&참가=>선발
(2) 국내&최신&기술&설치=>개발
(3) 게임&참가&인기&사용자&접속=>이벤트
(4) 운영&선발&경기&순위&규칙=>평가
(5) 게임&순위&이름=>스포츠
(6) 운영&스포츠&위원회&선수=>선발
(7) 게임&구성&선발&순위=>경기
(8) 게임&일정&선수&참가&운영=>스포츠
(9) 데이터&암호&통신망=>가입
(10) 게임&이용&문제=>규칙
(11) 그림&인기&서비스=>음악
(12) 그림&데이터&서비스=>엔진
(13) 데이터&프로그램=>음악
(14) 그림&데이터&프로그램=>사진
(15) 게임&설명&제공=>공략
(16) 게임&이용&기술=>개발
(17) 삭제&게임&개인전=>경고
(18) 게임&제공&일러스트=>설명

4.3 가중치가 부여된 연관 단어 지식 베이스

연관 단어 지식 베이스의 연관 단어에 가중치를 부여하기 위하여는 각 클래스의 훈련 문서 중 연관 단어 지식 베이스를 구축하기 위해 사용한 100개의 문서를 제외한 나머지 101번째부터 200번까지의 100개의 문서를 사용하여야 한다. Apriori 알고리즘은 각 클래스의 100개의 문서를 대상으로 신뢰도를 85로, 지지도를 0으로 지정함으로써 연관 단어를 마이닝할 수 있다. 마이닝 결과, Apriori 알고리즘은 총 250개의 연관 단어를 마이닝하였다. Naive Bayes 알고리즘은 이러한 결과를 식 (3)에 대입함으로써 연관 단어 지식 베이스의 연관 단어에 가중치를 추가한다. 표 4는 연관 단어 지식 베이스의 게임 클래스(*class1*)에 나타난 연관 단어에 가중치가 추가된 결과를 보인다.

4.4 Naive Bayes 분류자에 의한 문서의 분류

Naive Bayes 분류자는 웹문서 수집기에 의해 추출된 문서와 KTset95의 800개의 문서로 구성된 실험 문서를 식 (4)를 이용하여 가중치가 부여된 연관 단어 지식 베이스의 클래스의 하나로 분류한다. 표 5는 Naive Bayes 분류자가 실험 문서를 분류하는 예를 보인다. 이 예에서, 실험 문서는 {*게임&참가&인기&사용자&접속=>이벤트, 도메인&네트워크&계층=>호스트, 레이저&인크젯&플로터=>프린터, 게임&이용&기술=>개발, 운영&선발&경기&순위&*

표 4. 가중치가 부여된 연관 단어(게임 클래스)

연관 단어	nk	n	|AWKB|	가중치
게임&구성&선수&경기&스포츠&참가=>선발	23	25	231	0.09375
국내&최신&기술&설치=>개발	2	5	231	0.012712
게임&참가&인기&사용자&접속=>이벤트	25	28	231	0.100386
운영&선발&경기&순위&규칙=>평가	3	6	231	0.016878
게임&순위&이름=>스포츠	22	26	231	0.089494
운영&스포츠&위원회&선수=>선발	25	28	231	0.100386
게임&구성&선발&순위=>경기	21	23	231	0.086614
게임&일정&선수&참가&운영=>스포츠	23	27	231	0.093023
데이터&암호&통신망=>가입	1	4	231	0.008511
게임&이용&문제=>규칙	21	25	231	0.085938
그림&인기&서비스=>음악	1	3	231	0.008547
그림&데이터&서비스=>엔진	3	4	231	0.017021
데이터&프로그램=>음악	4	5	231	0.021186
그림&데이터&프로그램=>사진	5	6	231	0.025316
게임&설명&제공=>공략	24	29	231	0.096154
게임&이용&기술=>개발	25	28	231	0.100386
삭제&게임&개인전=>경고	25	27	231	0.100775
게임&제공&일러스트=>설명	21	26	231	0.085603

표 5. Naive Bayes 분류자에 의해 분류된 실험 문서

연관 단어	$class_1$	$class_2$	$class_3$	$class_4$	$class_5$	$class_6$	$class_7$	$class_8$
게임& 참가& 인기& 사용자& 접속=> 이벤트	1(0.100386)							
도메인& 네트워크& 계층=> 호스트						1(0.00635)		
레이저& 인크젯& 플로터=> 프린터								1(0.0321)
게임& 이용& 기술=> 개발	1(0.100386)							
운영& 선발& 경기& 순위& 규칙=> 평가	1(0.086614)							
$P(class_{ID})$	0.6					0.2		0.2
가중치곱	**0.0522**					0.0013		0.00642

규칙=> 평가 }의 연관 단어로 구성된다. Naive Bayes 분류자는 실험 문서를 대표하는 연관 단어의 가중치를 찾기 위해 연관 단어 지식 베이스를 참조한다. 이러한 결과를 식 (4)에 대입함으로써 실험 문서를 연관 단어 지식 베이스의 클래스로 분류할 수 있다. 결과적으로, 표 5에서는 실험 문서가 $class_1$의 게임 클래스로 분류됨을 보인다.

5. 성능 평가

본 논문에서는 제안된 연관 단어 지식 베이스를 기반으로 하는 가중치가 부여된 베이지안 문서 분류 방법(WBayesian-AWKB)의 성능을 평가하기 위해, 기존의 단순 베이지안 확률을 사용한 방법(Bayesian), 가중치가 부여된 베이지안 분류 방법(WBayesian), 상호 정보 계산에 의해 구축한 단어 사전을 기반으로 하는 베이지안 문서 분류 방법(WBayesian-VD)과 비교하였다. 이를 평가하기 위한 훈련 문서는 KTset95에 있는 1600개의 문서로 구성하고, 실험 문서는 웹문서 수집기에 의해 컴퓨터 분야의 URL로부터 수집된 800개의 웹문서와 KTset95에 있는 800개의 문서를 병합하여 구성한다. URL은 알타비스타, 야후 등의 기존의 정보 검색 엔진이 분류한 카테고리를 기준으로 선택한다. 또한 KTset95으로부터 선택할 실험 문서는 클래스별로 분류된 학습 문서에 있는 문서를 선택한다. 분류 성능을 평가하기 위해서 각 클래스로 분류된 문서를 대상으로 표 6과 같은 분할표를 작성한다[5].

표 6. 2x2-분할표

분류A \ 분류B		분할 B (웹문서:기존의 정보 검색 엔진에 의한 분류 KTset95:학습 문서로부터 추출)	
		YES	NO
분할 A (4가지 방법으로 생성된 분류)	YES	a	b
	NO	c	d

분류의 측정은 식 (5)의 F-measure 측정식을 이용한다. 식 (5)에서 P는 정확도, R은 재현율을 의미하며, 이 경우 F-measure의 값이 클수록 분류가 우수함을 의미한다. 여기서, β는 정확도에 대한 재현율의 상대적인 가중치를 나타내는 수치로, 1.0일 경우 정확도와 재현율의 가중치가 같다.

$$F_measure = \frac{(\beta^2+1)PR}{\beta^2 P+R} \quad P=\frac{a}{a+b}100\% \quad R=\frac{a}{a+c}100 \qquad (5)$$

본 실험에서는 β의 값을 1.0로 설정하여 분류 결과를 분석하였으며, 또한 β의 값을 0.5에서 1.4로 변화시키면서 F-measure의 결과 차이를 살펴보았다. 표 7은 정확도와 재현율을 식 (5)에 대입하여 분석한 결과를 나타낸다.

그림 6과 그림 7은 표 7의 결과를 바탕으로 한 재현율과 정확도의 성능 곡선을 나타낸다. 그림 7에서는 WBayesian-AWKB 방법의 재현율이 WBayesian-VD방법보다 0.44%, WBayesian방법보다는 2.59%, Bayesian방법보다는 3.32% 높음을 나타낸다.

그림 7에서는 WBayesian-AWKB 방법의 정확도가 WBayesian-VD방법보다 2.84%, WBayesian방법보다는 4.46%, Bayesian방법보다는 8.61% 높음을 나타낸다.

그림 8에서 β=1.0일 경우, WBayesian-AWKB 방법은 WBayesian-VD방법보다 1.63%, WBayesian 방법보다는 3.52%, Bayesian방법보다는 5.59% 높음을 나타낸다.

표 7. WBayesian-AWKB, WBayesian-VD, WBayesian, Bayesian의 성능 비교표

	WBayesian-AWKB			WBayesian-VD			WBayesian			Bayesian		
클래스	재현율 (%)	정확도 (%)	F-measure (%)	재현율 (%)	정확도 (%)	F-measure (%)	재현율 (%)	정확도 (%)	F-measure (%)	재현율 (%)	정확도 (%)	F-measure (%)
1	87.50	89.09	88.29	87.16	87.16	87.16	85.59	84.82	85.20	85.59	79.83	82.61
2	89.90	89.00	89.45	89.58	86.00	87.76	87.76	82.69	85.15	86.87	78.18	82.30
3	86.61	88.71	87.65	86.40	85.71	86.06	83.72	84.38	84.05	83.08	80.60	81.82
4	87.60	88.33	87.97	87.07	84.87	85.96	83.47	82.79	83.13	82.79	78.91	80.80
5	87.30	87.30	87.30	86.67	85.25	85.95	83.87	84.55	84.21	82.54	80.62	81.57
6	88.44	93.53	90.91	87.94	89.86	88.89	86.71	88.57	87.63	85.52	84.35	84.93
7	85.71	91.60	88.56	85.07	88.37	86.69	82.61	87.02	84.76	82.01	82.61	82.31
8	90.27	93.82	92.01	89.89	91.43	90.65	88.89	90.91	89.89	88.40	87.43	87.91
평균	87.92	90.17	89.02	87.47	87.33	87.39	85.33	85.72	85.50	84.60	81.57	83.03
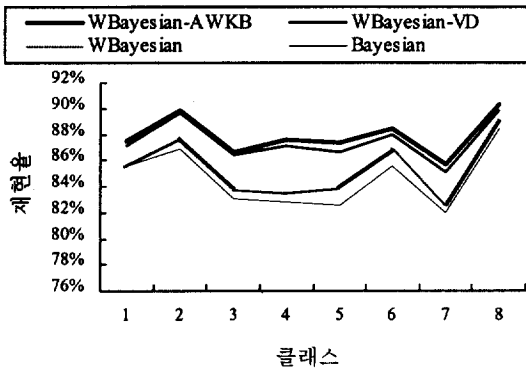
그림 6. WBayesian-AWKB, WBayesian-VD, WBayesian, Bayesian방법의 문서 분류 재현율
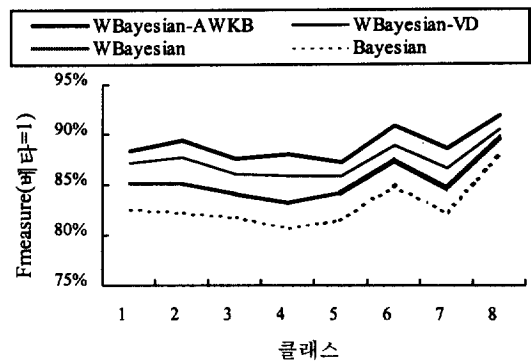
그림 7. WBayesian-AWKB, WBayesian-VD, WBayesian, Bayesian방법의 문서 분류 정확도
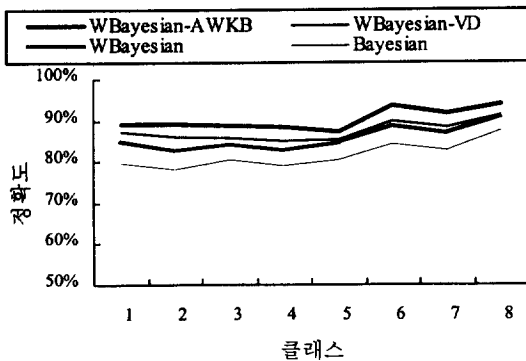
그림 8. F-measure에 의한 클래스별 문서 분류 성능 평가

그림 9. β의 변화에 따른 F-measure에 의한 클래스별 문서 분류 성능 평가

그림 9는 β값을 0.5에서 1.4로 변화시킴에 따른 F-measure의 성능 분석을 나타낸다. WBayesian-AWKB방법 뿐 아니라 WBayesian-VD방법과 WBayesian방법도 β값이 변할지라도 F-measure의 값은 일정한 값을 유지하므로 재현율과 정확도의 면에

서 비슷한 성능을 나타낸다. 그러나 Bayesian방법은 정확도보다는 재현율에서 더 높은 성능을 나타낸다. 평균적으로, WBayesian-AWKB방법은 WBayesian-VD방법보다 0.87%, WBayesian방법보다는 2.77%, Bayesian방법보다는 5.09% 높은 성능 차이를 보였다.

전체적으로 가중치를 부여한 연관 단어 지식 베이스나 상호 정보 계산에 의한 단어 사전을 사용한 분류 방법이 가중치만 부여한 방법이나 단순 베이지안 분류 방법보다는 성능이 우수함을 알 수 있다. 특히, 연관 단어 지식 베이스를 기반으로 하는 가중치가 부여된 베이지안 분류 방법은 가장 성능이 우수함을 나타냈다.

6. 결 론

본 논문에서는 기존의 베이지안 문서 분류 방법의 단점을 해결하기 위해, Apriori 알고리즘에 의한 연관 단어 지식 베이스를 기반으로 하는 가중치가 부여된 베이지안 문서 분류 방법을 제안하였다.

본 논문에서 제안한 방법은 두 가지의 장점을 갖는다. 첫째는 Naive Bayes 분류자가 정확한 분류를 가능하도록 연관 단어 지식 베이스를 구축했다는 것이다. 둘째는 실험 문서를 연관 단어의 집합으로 표현함으로써 단어 의미 중의성이라는 문제를 해결한 점이다. 본 논문에서는 제안된 분류 방법의 성능을 평가하기 위해, 기존의 단순 베이지안 분류 방법, 가중치가 부여된 베이지안 분류 방법, 상호 정보 계산에 의한 단어 사전을 이용한 가중치가 부여된 베이지안 문서 분류 방법과 비교하였다. 그 결과, 본 논문에서 제안된 방법이 상호 정보 계산에 의한 단어 사전을 이용하는 가중치가 부여된 베이지안 분류 방법보다는 0.87%, 가중치가 부여된 베이지안 분류 방법보다는 2.77%, 단순 베이지안 방법보다는 5.09% 높은 성능 차이를 보였다.

향후, 문서의 특징을 단순 명사가 아닌 복합 명사로 추출하여 연관 단어 지식 베이스를 기반으로 하는 베이지안 문서 분류 방법에 적용한다면 문서 분류의 성능이 보다 높아질 것이다.

참 고 문 헌

[1] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB Conference, Santiago, Chile, 1994.

[2] R. Agrawal and T. Imielinski and A. Swami, "Mining association rules between sets of items in large databases," Proceedings of the 1993 ACM SIGMOD Conference, Washington DC, USA, 1993.

[3] W. Frakes and R. Baeza-Yates, *Information Retrieval*, Prentice Hall, 1992.

[4] T. Joachims, "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization," ICML-97, 1997.

[5] V. Hatzivassiloglou and K. McKeown, "Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning," Proceedings of the 31st Annual Meeting of the ACL, pp. 172-182, 1993.

[6] K. Hyun-Jin and P. Jay-Duke and J. Myung-Gil and P. Dong-In, "Clustering Korean Nouns Based On Syntactic Relations and Corpus Data," Proceedings of the LASTED International Conference Artificial Intelligence and Soft Computing, 1998.

[7] M. Iwayama and T. Tokunaga, Cluster-Based Text "Categorization : A Comparison of Category Search Strategies," Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, 1995.

[8] K. Lang, "Newsweeder : Learning to filter netnews. In Prieditis and Russell(Eds)," Proceedings of the 21th International Conference on Machine Learning, pp. 331-339, 1995.

[9] D. D. Lewis, "Naive (Bayes) at forty: The Independence Assumption in Information Retrieval," European Conference on Machine Learning, 1998.

[10] D. D. Lewis, Representation and Learning in information retrieval, ph.D.thesis, Dept. of Computer and Information Science, University of Massachusetts, 1992.

[11] Y. H. Li and A. K. Jain, "Classification of Text Documents," The Computer Journal, Vol. 41, No. 8, 1998.
[12] M. E. Maron, "Automatic indexing : An experimental inquiry," Journal of the Association for Computing Machinery, pp. 404-417, 1961.
[13] T. Michael, *Maching Learning*, McGraw-Hill, pp. 154-200, 1997.
[14] A. McCallum and K. Nigram, "A Comparison of Event Models for Naive Bayes Text Classification," AAAI-98 Workshop on Learning for Text Categorization, 1998.
[15] H. Ragas and Cornelis H.A. Koster, "Four text classification algorithms compared on a Dutch corpus," Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 369-370, 1998.
[16] P. C. Wong and P. Whitney and J. Thomas, "Visualizing Association Rules for Text Mining," Proceedings of the 1999 IEEE Symposium on Information Visualization, pp. 120-123, 1999.
[17] O. Zamir and O. Etzioni, "Web Document Clustering: A Feasibility Demonstration," Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, 1998.
[18] 허준희, 가중치가 부여된 베이지안 분류자와 단어 군집을 이용한 한국어 문서 자동 분류, 인하대학교 대학원 컴퓨터공학과 석사학위 논문, 2000.

고 수 정

1990년 인하대학교 전자계산학과 졸업(학사)
1997년 인하대학교 교육대학원 전자계산교육(교육석사)
2000년 인하대학교 대학원 박사과정 수료
관심분야 : 데이터마이닝, 정보검색, 기계학습

이 정 현

1977년 인하대학교 전자공학과 졸업
1980년 인하대학교 대학원 전자공학과(공학석사)
1988년 인하대학교 대학원 전자공학과(공학박사)
1979년~1981년 한국전자기술연구소 시스템 연구원
1984년~1989년 경기대학교 전자계산학과 교수
1989년~현재 인하대학교 전자계산공학과 교수
관심분야 : 자연언어처리, HCI, 정보검색, 음성인식, 음성합성, 계산기 구조